Dynamic Negative Example Construction for Grammatical Error Correction using Contrastive Learning

Junyi He, Junbin Zhuang, and **Xia Li**[*]
Guangzhou Key Laboratory of Multilingual Intelligent Processing,
School of Information Science and Technology,
Guangdong University of Foreign Studies, Guangzhou, China
{junyihe,junbinzhuang,xiali}@gdufs.edu.cn

Abstract

Grammatical error correction (GEC) aims at correcting texts with different types of grammatical errors into natural and correct forms. Due to the difference of error type distribution and error density, current grammatical error correction systems may over-correct writings and produce a low precision. To address this issue, in this paper, we propose a dynamic negative example construction method for grammatical error correction using contrastive learning. The proposed method can construct sufficient negative examples with diverse grammatical errors, and can be dynamically used during model training. The constructed negative examples are beneficial for the GEC model to correct sentences precisely and suppress the model from over-correction. Experimental results show that our proposed method enhances model precision, proving the effectiveness of our method.

1 Introduction

Grammatical error correction (GEC) (Chollampatt and Ng, 2018; Kaneko et al., 2020; Kiyono et al., 2019) aims at correcting texts with different types of grammatical errors into natural and correct forms. It is an important research topic for both natural language processing and language education.

Most of the current GEC systems are developed for correcting writings by learners of English as a second language (Brockett et al., 2006; Chen et al., 2020a; Chollampatt and Ng, 2018; Kaneko et al., 2020; Kiyono et al., 2019). However, GEC for native writings is also worth exploring, as texts written by native speakers may also contain grammatical errors that should be corrected for enhancement of writing quality. Currently, it is not feasible to train a GEC model specifically for correcting native writings because GEC data containing native writings are not sufficient. Therefore, native writings are often corrected by GEC models that are trained on GEC data consisting of writings by non-native speakers such as the Lang-8 (Mizumoto et al., 2011) and NUCLE (Dahlmeier et al., 2013) datasets. However, the error type distribution, error density and fluency are inconsistent between the writings by non-native and native speakers. Therefore, those GEC models may over correct sentences and produce a low precision of error correction (Flachs et al., 2020). In terms of this issue, contrastive learning (CL) (Chen et al., 2020b; Chen et al., 2020c; Gao et al., 2021; Liu and Liu, 2021) can be incorporated to help alleviate the over correction behaviour of the GEC models. The core idea is to take the over-corrected sentences as negative examples, and to effectively avoid or alleviate the problem of over correction by increasing the distance between the anchor sentence and the negative examples. So the focus is on how to construct effective negative examples for training the GEC models effectively. Previous studies about GEC models mainly focus on data augmentation for generating pseudo parallel training pairs as complement for the current insufficient GEC training data (Zhao et al., 2019; Takahashi et al., 2020), or focus on improving the correction performance with a variety of model architectures (Awasthi et al., 2019; Stahlberg and Kumar, 2020; Sun and Wang, 2022), few of them focus on improving the performance of the GEC models with contrastive learning. To the best of our knowledge, Cao et al. (2021) is the only recent work

[*]Corresponding author: xiali@gdufs.edu.cn

for that. In their work, they propose two approaches for constructing negative examples. First, they treat the beam search candidates produced by an off-the-shelf GEC model as negative examples. They find that many output candidates generated by beam search contain erroneous edits, and the constructed negative examples help suppress the trained GEC model from producing erroneous edits. Second, the source sentence is treated directly as a negative example if it contains grammatical errors. Their intuition is that there should be differences between the corrected output sentence and the source sentence, otherwise the GEC model fails to detect any grammatical errors in the source sentence. The negative examples constructed in this way suppress the trained GEC model from outputting the erroneous source sentences as they are without any modifications.

Although the aforementioned study produces good performance, we believe that there are still two points that can be improved: (1) The negative examples constructed with beam search may not be sufficient. Many beam search output candidates are the same as the target sentence and cannot be used as negative examples. That leads to a small number of the generated negative examples. In addition, the beam search candidates may contain unrealistic grammatical errors with a small number of error types, limiting the diversity of grammatical errors in the generated negative examples. As a result, the low diversity of the negative examples makes the GEC model less easier to learn to distinguish the negative examples from the anchor, which limits the improvement of error correction performance brought by contrastive learning. (2) They construct the negative examples with their negative example construction methods before model training. As a result, the GEC model can only be able to see a fixed set of negative examples in each iteration during training, which may reduce the generalization ability of the GEC model.

To this end, we propose a dynamic negative example construction method for grammatical error correction using contrastive learning. The proposed method contains a negative example construction strategy that makes use of realistic grammatical error patterns produced by humans to generate sufficient negative examples with more diverse grammatical errors. With the constructed negative examples, the GEC model can learn to correct sentences precisely and be suppressed from over-correction. Moreover, the proposed strategy is simple and lightweight, enabling it to be applied dynamically during the training process. In this manner, our method enhances the generalization ability of the GEC model.

The main contributions of this work are as follows:

(1) We propose a dynamic negative example construction method for grammatical error correction using contrastive learning. The proposed method can construct sufficient negative examples with diverse grammatical errors, and can be dynamically applied during model training. The constructed negative examples are beneficial for the model to correct sentences precisely and suppress it from over-correction.

(2) We conduct extensive experiments on the public CWEB dataset that contains native writings, and compare our proposed method with existing GEC studies focusing on negative example construction. Experimental results show that our proposed method indeed enhances model precision and suppresses the GEC model from over-correction.

2 Related Work

In this section, we briefly review different GEC methods, including the early rule-based methods, the widely used methods based on machine translation or BERT, and the recently proposed GEC methods using contrastive learning.

GEC Methods based on Rules. Early GEC models are mostly rule-based pattern recognizers or dictionary-based linguistic analysis engines (Macdonald, 1983; Richardson and Braden-Harder, 1988; Sidorov et al., 2013; Sidorov, 2013). These rule-based methods require a set of pre-defined error recognition rules to detect grammatical errors in the input sentences. Once a certain span in the input sentence is matched by a certain rule, the error correction system provides a correction for the matched error.

GEC Methods based on Machine Translation. GEC models based on machine translation have been proposed to "translate" wrong sentences into correct sentences. Brockett et al. (2006) use a noisy channel model in conjunction with a statistical machine translation model for error correction. Felice et al. (2014) propose a hybrid GEC model that integrates grammatical rules and a statistical machine translation model. They also adopt some techniques such as type filtering. Zheng and Ted (2016) apply a neural machine translation model with the attention mechanism to GEC. In addition, they also introduce a method that uses a combination of an unsupervised alignment model and a word-level translation model to solve the problem of sparse and unrecognized words. Chollampatt et al. (2018) integrate four convolutional neural translation models combined with a re-scoring mechanism. Kiyono et al. (2019) construct the GEC model with Transformer and use many data augmentation techniques.

GEC Methods based on BERT. Many studies have introduced the Transformer-based deep bidirectional language model BERT (Devlin et al., 2019) into GEC, hoping that the correction performance can be improved with the help of its rich language knowledge and deep textual understanding ability. Awasthi et al. (2019) modify the BERT structure to predict the edit operation of each word in the source sentence by sequence tagging. Then they apply the edit operations to the source sentence to construct its correct form. Chen et al. (2020a) use BERT to predict and annotate error spans in the input sentence, and subsequently rewrite the annotated error spans with a sequence model. Kaneko et al. (2020) first fine-tune BERT with GEC data, then feed the output representations of the fine-tuned BERT to the GEC model as additional information for error correction.

GEC Methods based on Contrastive Learning. Contrastive learning (Chen et al., 2020b; Chen et al., 2020c; Gao et al., 2021; Liu and Liu, 2021) is a discriminative self-supervised learning method used to enhance the feature representation ability of deep learning models. For any training example, contrastive learning requires automatic construction of examples that are similar (positive examples) and dissimilar (negative examples) to an anchor. And during training, the model needs to reduce the distance between the anchor and the positive examples in the representation space, while to increase the distance between the anchor and the negative examples.

Cao et al. (2021) try to use contrastive learning to improve the error correction ability of the GEC model. Since constructing positive examples is difficult for GEC, they propose a margin-based contrastive loss, which only requires to construct negative examples and does not requires to construct positive examples. Their work is the most similar to ours. In view of the limitations of their negative example construction method, we propose a dynamic negative example construction method to better address the over correction problem of the GEC model in low error density native writings.

3 Our Method

In this section, we first describe the overall architecture of our method in Section 3.1. Then, we will detail the proposed negative example construction strategy in Section 3.2, and the proposed dynamic mechanism in Section 3.3.

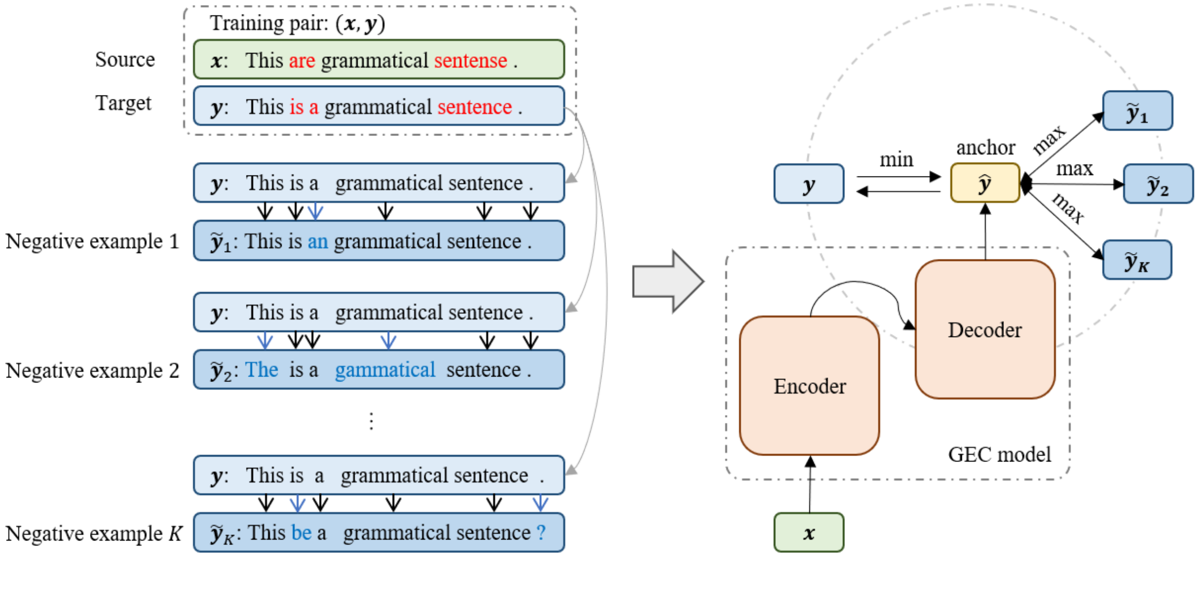
3.1 Overall Architecture

As mentioned above, the purpose of this work is to incorporate contrastive learning into the GEC model to effectively alleviate the problem of over correction by increasing the distance between the anchor sentence and the negative examples. We illustrate our method in Figure 1, which consists of two components: the negative example construction component and the contrastive learning component.

Given a training pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$ indicates the source sentence that may contain grammatical errors, x_i is the i^{th} word, m is the source sentence length, and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ indicates the target sentence that is grammatically correct, y_j is the j^{th} word, n is the target sentence length. The goal of the GEC task is to correct sentence $\boldsymbol{x}$ into sentence $\boldsymbol{y}$.

For the negative example construction component, we use the proposed negative example construction strategy to construct K negative examples $\tilde{\boldsymbol{Y}} = \{\tilde{\boldsymbol{y}}_{\boldsymbol{1}}, \tilde{\boldsymbol{y}}_{\boldsymbol{2}}, \cdots, \tilde{\boldsymbol{y}}_{\boldsymbol{K}}\}$ for the training pair $(\boldsymbol{x}, \boldsymbol{y})$. Each negative example $\tilde{\boldsymbol{y}}_{\boldsymbol{k}}$ is constructed as follows. First, several words in the target sentence $\boldsymbol{y}$ are randomly selected by a noising probability p. For each selected word y_j, a noised word y'_j is generated by the negative example construction strategy. The generated noised word y'_j is used to replace the selected word y_j. After replacing all selected words, the modified target sentence is treated as a constructed negative example $\tilde{\boldsymbol{y}}_{\boldsymbol{k}}$.

For the contrastive learning component, we first input the source sentence $\boldsymbol{x}$ into the GEC model and


Training pair: $(\boldsymbol{x}, \boldsymbol{y})$
Source
Target
$\boldsymbol{x}$: This are grammatical sentense .
$\boldsymbol{y}$: This is a grammatical sentence .
$\boldsymbol{y}$: This is a grammatical sentence .
Negative example 1
$\tilde{\boldsymbol{y}}_1$: This is an grammatical sentence .
$\boldsymbol{y}$: This is a grammatical sentence .
Negative example 2
$\tilde{\boldsymbol{y}}_2$: The is a gammatical sentence .
⋮
$\boldsymbol{y}$: This is a grammatical sentence .
Negative example K
$\tilde{\boldsymbol{y}}_K$: This be a grammatical sentence ?
$\boldsymbol{y}$
min
anchor
$\hat{\boldsymbol{y}}$
max
max
max
$\tilde{\boldsymbol{y}}_1$
$\tilde{\boldsymbol{y}}_2$
$\tilde{\boldsymbol{y}}_K$
Encoder
Decoder
GEC model
$\boldsymbol{x}$

Negative Example Construction **Contrastive Learning**

Figure 1: Overall architecture of our proposed method. Our method consists of two components: the negative example construction component and the contrastive learning component. For the negative example construction component, we use the proposed negative example construction strategy to construct K negative examples $\tilde{\boldsymbol{Y}} = \{\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \cdots, \tilde{\boldsymbol{y}}_K\}$ for the training pair $(\boldsymbol{x}, \boldsymbol{y})$. For the contrastive learning component, we first input the source sentence $\boldsymbol{x}$ into the GEC model and obtain the decoder output $\hat{\boldsymbol{y}}$. Then, we treat the decoder output $\hat{\boldsymbol{y}}$ as an anchor, and maximize the distance between the anchor $\hat{\boldsymbol{y}}$ and the negative examples $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \cdots, \tilde{\boldsymbol{y}}_K$ constructed by our proposed negative example construction strategy.

obtain the decoder output $\hat{\boldsymbol{y}}$. Then, we treat the decoder output $\hat{\boldsymbol{y}}$ as an anchor, and maximize the distance between the anchor $\hat{\boldsymbol{y}}$ and the negative examples $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \cdots, \tilde{\boldsymbol{y}}_K$ constructed by our proposed negative example construction strategy.

3.2 Negative Example Construction Strategy

In this section, we detail the proposed negative example construction strategy. It contains three schemes: realistic scheme, random scheme and linguistic scheme. In most cases, we use the realistic scheme for constructing negative examples. When the realistic scheme is not applicable, we use the random scheme or the linguistic scheme instead. We demonstrate examples using our proposed strategy in Table 1.

The realistic scheme makes use of the realistic grammatical error patterns produced by human beings for constructing negative examples. Realistic grammatical error patterns are effective for introducing realistic grammatical errors into error-free text and have been used in previous GEC studies for data augmentation (Li and He, 2021) and error-aware BERT fine-tuning (He et al., 2021). In this study, we utilize them for generating sufficient negative examples with more diverse grammatical errors.

Specifically, we first extract all realistic grammatical error patterns {WRONG:CORRECT} from the training data, where WRONG indicates an erroneous word in a sentence and CORRECT indicates its correction. Then, we reverse their key-value pairs into the form of {CORRECT:WRONG} for negative example construction. When constructing a negative example, for each word y_j selected from the target sentence $\boldsymbol{y}$, we randomly choose one of the {CORRECT:WRONG} patterns whose key CORRECT is y_j, and use the value WRONG as the noised word of y_j for replacement. The intuition is that we replace a correct word with a wrong word.

In practice, however, a word y_j randomly selected from the target sentence may not be one of the keys in the available {CORRECT:WRONG} pairs, such as an out-of-vocabulary word. To handle this case, we

Target sentence $\boldsymbol{y}$	We are exploring negative example construction strategies.
Realistic Scheme	We is exploring negative example construction strategy.
Random Scheme	We are exploring Title example fill strategies.
Linguistic Scheme	
-synonym	We are exploring passive example building strategies.
-inflection	We are explored negative example constructed strategies.
-function word	They are exploring negative example construction strategies.
-case	we are exploring Negative example construction strategies.
-misspelling	We air exploding negative example construction strategies.

Table 1: Demonstration of our proposed negative example construction strategy, which contains three schemes. Note that each linguistic transformation in the linguistic scheme is demonstrated separately for clarity. In practice, one of the linguistic transformations will be randomly selected.

propose two additional schemes as compromise:

1) Random Scheme. Such a particular word is replaced by another word sampled from the vocabulary of the dataset in a uniform distribution.

2) Linguistic Scheme. Such a particular word is replaced by one of the five linguistic transformations described below. These linguistic transformations are used by some GEC studies to mimic realistic grammatical errors (Takahashi et al., 2020; Li and He, 2021; He et al., 2021).

○ *Synonym Transformation.* Replacing y_j with one of its synonyms. It is helpful for generating word misuse errors (noun errors, verb errors, adjective errors, etc.) commonly appeared in writings, such as misusing *"situation"* as *"condition"*.

○ *Inflection Transformation.* Replacing y_j with one of its inflections. It imitates inflection misuse in the writings, such as misusing noun declension and verb conjugation. E.g., using the present tense of *"is"* where the past tense *"was"* is required.

○ *Function Word Transformation.* Replacing y_j with another function word that belongs to the same function word category of y_j. It imitates the improper function word uses in writings, such as misusing *"at"* as *"in"* and misusing *"to"* as *"towards"*.

○ *Case Transformation.* Replacing y_j with one of the three case patterns: lowercase, uppercase, and capitalize. It mimics the case errors made frequently by native English speakers due to their carelessness, such as lower-casing country names, city names and abbreviations.

○ *Misspelling Transformation.* Replacing y_j with one of its 10 most similar words. It mimics the misspelling errors commonly appeared in writings by the native English speakers due to carelessness or rapid typing with keyboards.

3.3 Dynamic Construction

Many contrastive learning studies (Chen et al., 2020b; Chen et al., 2020c; Gao et al., 2021) have proved that the variety of negative examples is beneficial for improving the performance of the trained model. In our method, the proposed negative example construction strategy is based on rules, and the operations required for constructing negative examples are merely random sampling and replacement. Therefore, they are lightweight and consume little time, enabling them to be dynamically applied during the training process.

As shown in Figure 2, we depict the proposed dynamic negative example construction, and compare it with the static negative example construction. In the figure, $(\boldsymbol{x}, \boldsymbol{y})$ denotes a training pair, where $\boldsymbol{x}$ denotes the source sentence and $\boldsymbol{y}$ denotes the target sentence. $\tilde{\boldsymbol{Y}}$ denotes the constructed negative examples. f_{static} denotes the static negative example construction strategy and f_{dynamic} denotes our proposed dynamic negative example construction strategy.

With static construction (the higher part of the figure), the negative examples $\tilde{\boldsymbol{Y}}$ (blue) for the training pair $(\boldsymbol{x}, \boldsymbol{y})$ are constructed before the training process. And during training, the same set of negative examples constructed ($\tilde{\boldsymbol{Y}}$) is used in each iteration. On the contrary, with dynamic construction (the lower part of the figure), different sets of negative examples are constructed dynamically for the training pair $(\boldsymbol{x}, \boldsymbol{y})$ in each iteration during training. Specifically, in iteration 1, a set of negative examples $\tilde{\boldsymbol{Y}}_1$ (orange) are constructed with the negative example construction strategy f_{dynamic}. Similarly, another set of negative examples $\tilde{\boldsymbol{Y}}_2$ (yellow) are constructed in iteration 2. In this manner, for the same training pair, dynamic construction enables the model to see different sets of negative examples in different iterations during training, and significantly increases the variety of the negative examples.

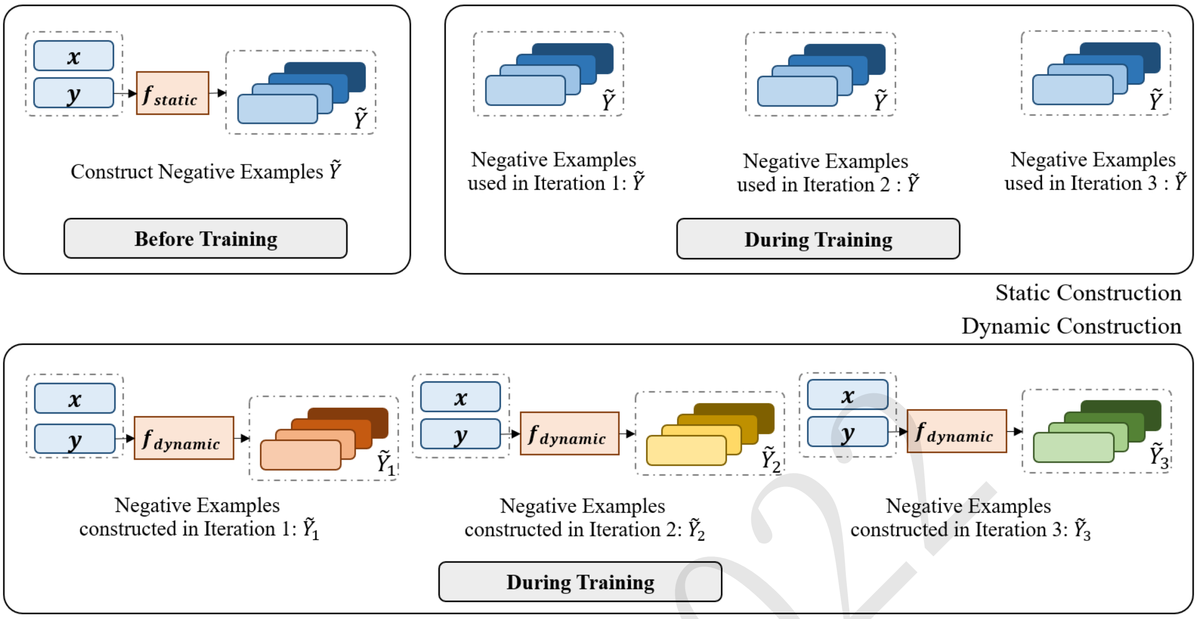

x
y
f static
Ỹ
Construct Negative Examples Ỹ
Before Training
Ỹ
Negative Examples used in Iteration 1 : Ỹ
Ỹ
Negative Examples used in Iteration 2 : Ỹ
Ỹ
Negative Examples used in Iteration 3 : Ỹ
During Training
Static Construction
Dynamic Construction
x
y
f dynamic
Ỹ1
Negative Examples constructed in Iteration 1: Ỹ1
x
y
f dynamic
Ỹ2
Negative Examples constructed in Iteration 2: Ỹ2
x
y
f dynamic
Ỹ3
Negative Examples constructed in Iteration 3: Ỹ3
During Training

Figure 2: Demonstration of the proposed dynamic negative example construction and its comparison with the static negative example construction.

3.4 Model Training

Following Cao et al. (2021), we use the weighted sum of the negative log likelihood loss L^{NLL} and a margin-based contrastive loss L^{CL} as the training loss L for each training pair $(\boldsymbol{x}, \boldsymbol{y})$ to optimize model parameters, as in Equation 1. α is a weighting parameter that controls the relative importance of the two losses. During training, the negative log likelihood loss L^{NLL} (Equation 2) increases the similarity between the model output $\hat{\boldsymbol{y}}$ and the target sentence $\boldsymbol{y}$. And the contrastive loss L^{CL} (Equation 3) discourages the model from generating each negative example $\tilde{\boldsymbol{y}}_{\boldsymbol{k}}$ that contains grammatical errors. K is the number of constructed negative examples, and γ is the margin.

$$L = \alpha \cdot L^{\text{NLL}} + (1-\alpha) \cdot L^{\text{CL}} \tag{1}$$

$$L^{\text{NLL}} = -\boldsymbol{y} \log \hat{\boldsymbol{y}} \tag{2}$$

$$L^{\text{CL}} = \frac{1}{K} \sum_{k=1}^{K} \max(-\boldsymbol{y} \log \hat{\boldsymbol{y}} + \tilde{\boldsymbol{y}}_{\boldsymbol{k}} \log \hat{\boldsymbol{y}} + \gamma, 0) \tag{3}$$

4 Experiments

4.1 Datasets

We use the CWEB dataset (Flachs et al., 2020) for experiments. It contains low error density writings from native English speakers and includes two domains. The G domain (CWEB-G) contains writings with a higher number of grammatical errors, and the S domain (CWEB-S) contains more professional writings with fewer grammatical errors.

The CWEB dataset only contains development data and test data but no training data. Following previous studies (Cao et al., 2021; Flachs et al., 2020), we extract the first 1,000 samples of CWEB-G and the first 1,000 of CWEB-S from the original development data and combine them to form the training data, which are used for training models and extracting realistic grammatical error patterns. The remaining of the original development data are taken as new development data for obtaining the best model during training. The original test data of CWEB are left unchanged, with which we evaluate the trained GEC models. We use ERRANT (Bryant et al., 2017) to calculate precision, recall and $F_{0.5}$ for evaluating the correction performance of the GEC models. Statistics of the dataset are shown in Table 2.

The grammatical errors and their corresponding corrections are annotated by two annotators. When training the model, we only use the corrections from annotator 1 as target sentences. When evaluating the trained model on test data, we calculate the scoring performance against each annotator and take the average for report.

Splits	**Original**		**Derived**	
Train	-	-	2,867	1,862
Dev	3,867	2,862	1,000	1,000
Test	3,981	2,864	3,981	2,864

Table 2: Statistics of the CWEB dataset. **Original** is the statistics of the original dataset and **Derived** is the statistics after splitting the development set into training and development data.

4.2 Experiment Settings

We use Transformer-big (Vaswani et al., 2017) as the model architecture. Following Cao et al. (2021), we use the pre-trained weights of GEC-PD (Kiyono et al., 2019) to initialize the GEC model. We use the Adam (Kingma and Ba, 2014) optimizer with the learning rate set to 3e-5. We train the model for 10 epochs and validate it after each epoch on the development set. Model weights of the smallest validation loss is used as the best model for evaluation on the test set. We construct $K = 4$ negative examples for each training pair and set the noising probability p to 0.15. We run 3 times with different seeds for each experiment and take the average of the 3 runs for report to reduce randomness.

4.3 Compared Models

We compare our method with several strong baselines to prove the effectiveness of the proposed method:

Direct Inference. Making predictions on CWEB test data directly with an off-the-shelf GEC model developed for correcting writings by learners of English as a second language, without further training on CWEB training data. In experiments, we use GEC-PD (Kiyono et al., 2019) for this purpose.

NLL. The model is first initialized with the weights of the GEC-PD model. Then, it is trained on the training data merely with negative log-likelihood (i.e., without contrastive learning) and evaluated on the test data.

$\mathbf{CL_{2021}}$. The model propsoed by Cao et al. (2021). They first initialize the model with the weights of GEC-PD. Then, they train the model on the training data with their contrastive learning method, and evaluate the trained model on the test data.

4.4 Overall Results and Analysis

The overall experimental results are shown in Table 3. **Direct Inference** are the results by the GEC-PD (Kiyono et al., 2019) model without further training on the training set. **NLL** are the results of the GEC model initialized with the weights of GEC-PD and trained merely using the negative log-likelihood loss without contrastive learning. $\mathbf{CL_{2021}}$ are the results reported in the paper of Cao et al. (2021). **Ours (Realistic+Rand)** are the results of our proposed method with realistic scheme & random scheme, and **Ours (Realistic+Ling)** are the results of our proposed method with realistic scheme & linguistic scheme. **Average** are the average results of CWEB-G and CWEB-S. The best scores of each column are shown in bold.

Model	CWEB-G			CWEB-S			Average		
	P	**R**	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$
Direct Inference	21.18	23.01	21.45	17.27	15.76	16.92	19.22	19.38	19.18
NLL	40.46	18.93	32.76	36.78	16.66	29.51	38.62	17.79	31.14
$\mathbf{CL_{2021}}$	37.21	**23.15**	33.03	36.30	**20.40**	**31.34**	36.76	**21.78**	32.19
Ours (Realistic+Rand)	41.37	19.80	33.80	38.06	17.08	30.48	39.71	18.44	32.14
Ours (Realistic+Ling)	**42.42**	19.06	**33.89**	**39.10**	16.80	30.82	**40.76**	17.93	**32.36**

Table 3: Overall experiment results. **Direct Inference** are the results by the GEC-PD (Kiyono et al., 2019) model without further training on the training set. **NLL** are the results of the GEC model initialized with the weights of GEC-PD and trained merely using the negative log-likelihood loss without contrastive learning. $\mathbf{CL_{2021}}$ are the results reported in the paper of Cao et al. (2021). **Ours (Realistic+Rand)** are the results of our proposed method with realistic scheme & random scheme, and **Ours (Realistic+Ling)** are the results of our proposed method with realistic scheme & linguistic scheme. **Average** are the average results of CWEB-G and CWEB-S. The best scores of each column are shown in bold.

First, it is shown that the results of **Direct Inference** with GEC-PD are low. It's average $F_{0.5}$ is 19.18. And its average precision is only 19.22, which is lower than other results by a large margin. That supports the finding that the GEC model developed for correcting writings by learners of English as a second language indeed produces low performance on the writings by native English speakers due to the low error density (Flachs et al., 2020).

Second, we find that after training GEC-PD on the CWEB training data with our proposed method, the results are improved. Specifically, the average $F_{0.5}$ of our proposed **Realistic+Ling** (32.36) is higher than **NLL** (31.14) by 1.22, and higher than $\mathbf{CL_{2021}}$ (32.19) by 0.17.

Third, we can also see that our method significantly boosts the precision of the GEC model. For example, the precision of **Realistic+Ling** in CWEB-G and CWEB-S are 42.42 and 39.10, which are 1.96 and 2.32 higher than **NLL**, 5.21 and 2.80 higher than $\mathbf{CL_{2021}}$. At the same time, it also produces the highest average precision (40.76). The higher precision of our GEC model illustrates that the grammatical errors detected by the model indeed are erroneous, rather than accurate. In the task of grammatical error correction, a GEC model with an ability to **accurately** correct the detected grammatical errors (higher precision) is more preferred than one with an ability to detect many grammatical errors but fail to correct them (higher recall). This is also reflected by the evaluation metric $F_{0.5}$ of GEC, which values the precision twice as the recall. Therefore, our proposed method is beneficial for enhancing the correction performance of the GEC model, as it indeed makes the model correct the detected grammatical errors precisely and suppresses the model from over-correction.

Finally, the results also show that **Realistic+Ling** produces higher average precision (40.76) and average $F_{0.5}$ (32.36) than **Realistic+Rand** (39.71 and 32.14). It proves that the pseudo grammatical errors generated by the linguistic transformations are beneficial and effective for the construction of negative examples, which leads to a better GEC model.

5 Discussion and Analysis

5.1 Case Study

In order to demonstrate that our proposed negative example construction strategy can indeed generate sufficient negative examples with realistic and diverse grammatical errors, we extracted one training pair from the CWEB dataset accompanied by their corresponding negative examples constructed by Cao et al. (2021)'s method and those constructed with our proposed method, as shown in Table 4. In the training pair, there is a case error (*"allow"* → *"Allow"*), which is coloured red. In the negative examples, noises introduced by the negative example construction methods are coloured blue.

We can see that the first, third and fourth negative examples constructed by Cao et al. (2021)'s method are the same as the source sentence. The second example contains an insertion error (*"pick"* → *"pick*

Source sentence	allow them to pick some coloring sheets that you can print for them.
Target sentence	Allow them to pick some coloring sheets that you can print for them.
Negative examples by Cao et al. (2021)	allow them to pick some coloring sheets that you can print for them.
	Allow them to pick up some coloring sheets that you can print for them.
	allow them to pick some coloring sheets that you can print for them.
	allow them to pick some coloring sheets that you can print for them.
Negative examples by our method	Allow them to pick some coloring sheets that you can print with them.
	Allow them to pick some coloring piece that you can prunt for them.
	Allow them to pick sum coloring sheets that you can print for them.
	allow them to pick some coloring sheets that you can print in them.

Table 4: Case study. We extracted one training pair from the CWEB dataset accompanied by their corresponding negative examples constructed by Cao et al. (2021)'s method and those constructed with our proposed method. Grammatical errors in the training pair are coloured red. Noises in the negative examples introduced by the negative example construction methods are coloured blue. The negative examples constructed with out proposed method are more sufficient with more diverse grammatical errors.

Method	**Static**			**Dynamic**		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Realistic+Rand	39.90	17.74	31.78	39.71	18.44	**32.14**
Realistic+Ling	38.75	18.73	31.77	40.76	17.93	**32.36**

Table 5: Scoring performance comparison of the proposed Realistic+Rand and Realistic+Ling methods with static and dynamic construction respectively.

up"). Obviously, these negative examples do not contain diverse and realistic grammatical errors, which is not helpful for the model to learn to correct properly from contrastive learning. On the other hand, the negative examples constructed using our proposed method contain a large number of diverse and realistic grammatical errors. For instance, the first example contains a preposition error (*"for"* → *"with"*). The second example contains a synonym error (*"sheets"* → *"piece"*) and a misspelling error (*"print"* → *"prunt"*). From the negative examples with diverse and realistic errors, the GEC model can better learn to correct sentences precisely through contrastive learning.

5.2 Effect of Dynamic Construction

As mentioned above, our proposed dynamic negative example construction can increase the variety of the negative examples during model training. In this section, we investigate the effect of dynamic construction by comparing the scoring performance of the proposed Realistic+Rand and Realistic+Ling methods with static and dynamic construction respectively.

The experimental results are shown in Table 5. The left half shows the results of static construction, while the right half shows the results of dynamic construction. The results of each negative example construction method are the average of CWEB-G and CWEB-S. The higher results between the static and dynamic construction of each method are bolded.

As shown in the table, the dynamic results are generally higher than the static results. Specifically, the $F_{0.5}$ of static Realistic+Rand is 31.78, while that of the dynamic one is 32.14, with a performance gap of 0.36. The $F_{0.5}$ of static Realistic+Ling is 31.77, while the dynamic one is 32.36, with a large performance gap of 0.59. It proves that by increasing the variety of the negative examples during training, dynamic construction indeed increases the variety of the negative examples, thereby avoiding overfitting and enhancing the generalization ability of the GEC model.

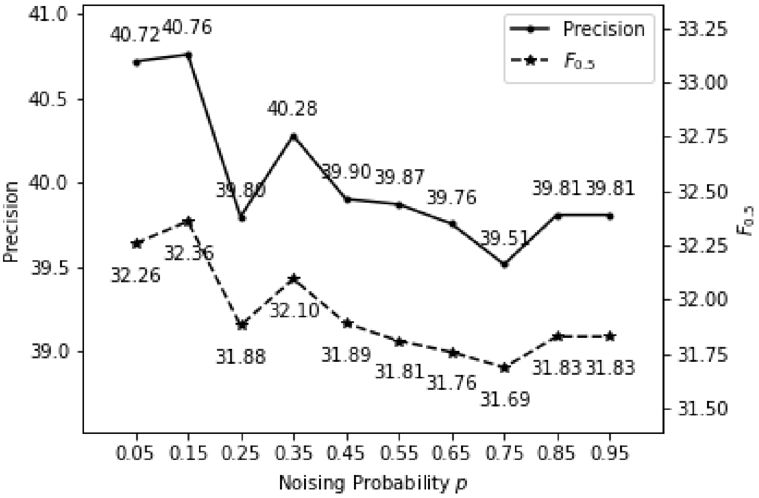


Figure 3: Scoring performance of the GEC model at different values of p, from 0.05 to 0.95 at a 0.1 interval. The dynamic Realistic+Ling strategy is used for constructing negative examples in the experiment. The experiment results are obtained from averaging the results of CWEB-G and CWEB-S. When p is set to 0.15, the scores reaches the highest (precision=40.76, $\mathrm{F}_{0.5}$=32.36). As p gradually increases, the precision and $\mathrm{F}_{0.5}$ drop gradually.

5.3 Effect of the Noising Probability

When constructing a negative example with our proposed negative example construction strategy, a noising probability p should be determined to randomly select words from the target sentence for replacement. In this section, we analyze the impact of different values of p on the correction performance. Specifically, we construct negative examples with the proposed dynamic Realistic+Ling strategy according to different values of p, from 0.05 to 0.95 at a 0.1 interval. The precision and $F_{0.5}$ at each probability are shown in Figure 3, which are obtained from averaging the results of CWEB-G and CWEB-S.

The results show that when p is set to 0.15, the scores reaches the highest (precision=40.76, $\mathrm{F}_{0.5}$=32.36). As p gradually increases, the precision and $\mathrm{F}_{0.5}$ drop gradually. The reason may be that as p increases, more words are selected from the target sentence for replacement. Therefore, the negative examples constructed are more different from the target sentence. The greater the difference between the target sentence and the negative example, the easier it is for the GEC model to compare their differences, and the smaller the improvement in the error correction ability of the model obtained from contrastive learning.

6 Conclusion

In this paper, a dynamic negative example construction method for grammatical error correction using contrastive learning is proposed. The proposed method constructs sufficient negative examples with diverse grammatical errors dynamically during model training. The constructed negative examples are beneficial for the GEC model to correct sentences precisely and suppress the model from over-correction. Experimental results show that our proposed method enhances the correction precision significantly. In this study, positive example construction strategy is not proposed for grammatical error correction using contrastive learning, as it is hard to construct sentences that are morphologically different from but semantically identical to the target sentence. One possible solution for that may be utilizing data augmentation. In future work, we will investigate this topic in depth.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 61976062).

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China, November. Association for Computational Linguistics.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia, July. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada, July. Association for Computational Linguistics.

Hannan Cao, Wenmian Yang, and Hwee Tou Ng. 2021. Grammatical error correction with contrastive learning in low error density domains. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4867–4874, Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020a. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169, Online, November. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020c. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland, June. Association for Computational Linguistics.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online, November. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Junyi He, Xia Li, Han Su, Xinyin Chen, Hao Yang, and Minghao Chen. 2021. Ea-mlm: Error-aware masked language modeling for grammatical error correction. In *2021 International Conference on Asian Language Processing (IALP)*, pages 363–368.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online, July. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China, November. Association for Computational Linguistics.

Xia Li and Junyi He. 2021. Data augmentation of incorporating real error patterns and linguistic knowledge for grammatical error correction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 223–233, Online, November. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online, August. Association for Computational Linguistics.

Nina H Macdonald. 1983. Human factors and behavioral science: The unix™ writer's workbench software: Rationale and design. *Bell System Technical Journal*, 62(6):1891–1908.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Stephen D. Richardson and Lisa C. Braden-Harder. 1988. The experience of developing a large-scale natural language text processing system: Critique. In *Second Conference on Applied Natural Language Processing*, pages 195–202, Austin, Texas, USA, February. Association for Computational Linguistics.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. 2013. Rule-based system for automatic grammar correction using syntactic n-grams for English language learning (L2). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–101, Sofia, Bulgaria, August. Association for Computational Linguistics.

Grigori Sidorov. 2013. Syntactic dependency based n-grams in rule based automatic english as second language grammar correction. *International Journal of Computational Linguistics and Applications*, 4(2):169–188.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online, November. Association for Computational Linguistics.

Xin Sun and Houfeng Wang. 2022. Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693, Dublin, Ireland, May. Association for Computational Linguistics.

Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online, July. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California, June. Association for Computational Linguistics.

Computational Linguistics

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota, June. Association for Computational Linguistics.